

Narration	Time
'Access Modifiers in PERL' پر اسپوکن ٹیوٹوریل میں خوش آمدید۔	00:01
اس ٹیوٹوریل میں ہم مندرجہ ذیل کے بارے میں سیکھیں گے: ویریبلس کا scope یعنی دائرہ کار 'Private' ویریبلس 'Dynamically scoped' ویریبلس 'Global' ویریبلس	00:07
اس ٹیوٹوریل کے لئے میں: 'ابنٹو لینکس' 12.04 آپریٹنگ سسٹم 'Perl 5.14.2' اور 'gedit' ٹیکسٹ ایڈیٹر استعمال کر رہا ہوں	00:19
آپ اپنی پسند کا کوئی بھی ٹیکسٹ ایڈیٹر استعمال کر سکتے ہیں۔	00:32
آپ کو 'Perl' پروگرامنگ کا بنیادی علم ہونا چاہئے۔	00:36
اگر نہیں تو اسپوکن ٹیوٹوریل ویب سائٹ پر متعلقہ 'Perl' اسپوکن ٹیوٹوریلز کو دیکھیں۔	00:40
اب 'ویریبلس' کے سکوپ کے تعارف سے شروع کرتے ہیں۔	00:47
ویریبل کا سکوپ کوڈ کا وہ علاقہ ہے جس سے ویریبل حاصل کیا جاسکتا ہے۔	00:51
دوسرے الفاظ میں، یہ ویریبلس کے اظہار کا حوالہ دیتا ہے۔	00:58
پہلے، ہم 'Perl' میں 'my'، 'local' اور 'our' موڈ فارز کے بارے میں بات کریں گے۔	01:03
'my' کا مطلب ہے 'Private variables'،	01:10
'local' کا مطلب ہے 'Dynamically scoped variables'،	01:13
'our' کا مطلب ہے 'Global variables'۔	01:17
'my' کی وارڈ کے ساتھ ڈکلیئر ہوئے ویریبلس کی اس block کے باہر سکوپ ختم ہو جائے گی جس میں وہ ڈکلیئر ہوئے ہیں۔	01:20
آپ ایک ویریبل کو ویلیو دیے بغیر ڈکلیئر کر سکتے ہیں، جیسے my \$fvalue semicolon	01:28

01:37	آپ ایک وریبل کو ویلیو تفویض کر کے بھی ڈکلیئر کر سکتے ہیں، جیسے
01:43	my \$ fValue = 1 semicolon
01:48	'my \$ fname' = ڈبل کوٹس میں 'Rahul' semicolon
01:55	ایک ہی 'my' اسٹیٹمنٹ کے ساتھ کئی وریبلز کو ڈکلیئر کرنے کے لئے سنیگیس مندرجہ ذیل ہے:
02:02	'my' بریکٹ کھولیں '\$ fname' کا '\$ lname' کا '\$ age' بریکٹ بند کریں semicolon
02:12	اب ایک سیمپل پروگرام استعمال کرتے ہوئے 'private' وریبلز کو سمجھتے ہیں۔
02:17	میرے پاس پہلے سے سیمپل پروگرام ہے۔ میں اسے 'gedit' ٹیکسٹ ایڈیٹر میں کھولتا ہوں۔
02:24	terminal کھولیں اور ٹائپ کریں: 'gedit scope' ہائفن my ڈاٹ 'pl ampersand' اور اینٹر دبائیں۔
02:34	'Scope-my dot pl' فائل اب 'gedit' میں کھلتی ہے۔
02:39	سکرین پر دکھائے طریقے سے مندرجہ ذیل کوڈ ٹائپ کریں۔ اب میں کوڈ سمجھاتا ہوں۔
02:46	یہاں، میں نے 'my' کی وارڈ کے ساتھ ایک 'private' وریبل '\$ fname' ڈکلیئر کیا ہے۔
02:52	اور اسے ویلیو 'Raghu' مقرر کیا ہے۔
02:56	اس بلاک میں، 'print' اسٹیٹمنٹ '\$ fname' وریبل کے ویلیو جو کہ 'Raghu' ہے، کو پرنٹ کرتا ہے۔
03:04	اگلے بلاک میں، میں نے اسی 'private' وریبل '\$ fname' کو 'Other' ویلیو مقرر کیا ہے۔
03:11	لہذا، 'پرنٹ اسٹیٹمنٹ' اس خصوصی بلاک میں 'Other' پرنٹ کرے گا۔
03:17	اس پروگرام میں آخری print statement کوئی آؤٹ پٹ پرنٹ نہیں کرے گا۔
03:23	ایسا اس لئے ہے کیونکہ پہلے سے وضاحت شدہ بلاکس کے سکوپ سے باہر، '\$ fname' کی کوئی ویلیو مقرر نہیں ہے۔
03:32	اب فائل سیو کرنے کے لئے 'Ctrl + S' دبائیں۔
03:37	اب پروگرام کو ایکزیکوٹ کرتے ہیں۔
03:40	terminal پر واپس جائیں اور ٹائپ کریں 'perl scope' ہائفن my ڈاٹ 'pl' اور اینٹر دبائیں۔

03:49	آؤٹ پٹ مندرجہ ذیل کی طرح لگتا ہے: 'Block 1: Raghu' 'Block 2: Other' 'Outside Block:' کوئی آؤٹ پٹ نہیں.
03:59	لہذا، 'my' ویریبل کے سکوپ تک رسائی کوڈ کے صرف ایک خاص بلاک میں حاصل ہوتا ہے.
04:06	اب، موجودہ پروگرام کو تھوڑا سا تبدیل کرتے ہیں.
04:10	اب آخری print اسٹیٹمنٹ سے پہلے بلاکس کے باہر my \$ fname = ڈبل کوٹس میں John semicolon جوڑتے ہیں. تبدیلیوں کو سیو کریں.
04:23	'ٹرمینل' پرواپس جائیں اور پہلے کی طرح ایکزیکوٹ کریں.
04:28	ظاہر آؤٹ پٹ کا تجزیہ کریں
04:32	امید ہے آپ بلاک میں اور بلاک سے باہر 'my' ویریبل کے استعمال کرنے کا سکوپ سمجھنے کے قابل ہو گئے ہیں.
04:41	آگے، ہم 'Perl' میں 'dynamically scoped variable' کے بارے میں دیکھیں گے.
04:47	'Local' کی وارڈ 'global' ویریبل کو ایک عارضی سکوپ فراہم کرتا ہے.
04:52	ویریبل حقیقی بلاک سے کال ہوئے کسی بھی فنکشن کے لئے دستیاب ہوتا ہے.
04:58	آپ مندرجہ ذیل کی طرح 'local' ویریبل ڈیکلئیر کر سکتے ہیں: 'local \$ fValue = 100 semicolon' 'local \$ fname' = 'Rakesh' semicolon
05:13	اب ایک سیمپل پروگرام استعمال کر کے اسے سمجھتے ہیں.
05:17	'ٹرمینل' کھولیں اور ٹائپ کریں : gedit scope 'ہائفن local ڈاٹ pl ampersand اور اینٹر دبائیں.
05:27	یہ 'gedit' میں 'scope' ہائفن local ڈاٹ 'pl' فائل ظاہر کرے گا.
05:33	سکرین پر دکھائے گئے طریقے سے مندرجہ ذیل کوڈ ٹائپ کریں. اب میں کوڈ سمجھاتا ہوں.
05:40	یہاں، پہلی لائن میں، ہم نے ایک ویریبل '\$ fname' ڈیکلئیر کیا ہے اور اسے انشلائز کیا ہے.
05:47	فنکشن 'Welcome ()' میں، ہم نے ایک 'local' ویریبل اسی نام '\$ fname' سے ڈیکلئیر کیا ہے.

05:54	ویریبل کے نام سے پہلے 'local' کی وارڈ پر توجہ دیں
05:59	اور ہم نے اس ویریبل کو ویلیو 'Rakesh' مقرر کیا ہے۔
06:03	تو بنیادی طور پر، فنکشن 'Welcome ()' میں، '\$ fname' ایک نئے عارضی local ویریبل کی طرح تبدیل ہوتا ہے۔ پھر، فنکشن 'Hello ()' کال کیا جاتا ہے۔
06:15	یہاں 'Hello ()' کی فنکشن ڈیفینیشن یعنی تعارف ہے۔
06:18	پروگرام کے آخر میں، ہم 'Welcome ()' اور 'Hello ()' دونوں فنکشنس کو کال کر رہے ہیں۔
06:25	اب پروگرام کو سیو کرنے کے لئے 'Ctrl + S' دبائیں۔
06:29	اب پروگرام کو ایکزیکوٹ کرتے ہیں۔
06:31	'ٹرمینل' پرواپس جائیں اور ٹائپ کریں : 'perl scope' ہانقن 'local.pl' اور اینٹر دبائیں۔
06:41	آؤٹ پٹ مندرجہ ذیل کی طرح لگتا ہے۔ "Hello, Rakesh"! "Hello, Welcome to Spoken tutorials!"
06:48	اب آؤٹ پٹ کو سمجھتے ہیں۔
06:51	جب فنکشن 'Welcome ()' کال ہوتا ہے، تو اس کے اندر فنکشن 'Hello ()'، 'local' ویریبل تک رسائی کرتا ہے۔
06:59	'Welcome ()' میں، '\$ fname'، ویلیو 'Rakesh' رکھتا ہے۔
07:04	اس کے بعد فنکشن 'Hello ()' ایک بار پھر ویریبل '\$ fname' کو رسائی کرتا ہے۔
07:11	لیکن اس وقت، یہ وہ ویریبل '\$ fname' ہے جو 'Welcome to spoken tutorials' کو انشلا نر ہوا تھا۔
07:19	اس فنکشن 'Welcome ()' میں لوکل ویریبل '\$ fname' کو رسائی نہیں کرتا ہے
07:25	جس کا مطلب ہے کہ بلاک 'Welcome ()' چھوڑنے کے بعد، لوکل ویریبل 'scope' کو پرانی پوزیشن پرواپس لاتا ہے۔
07:32	آگے، ہم 'Perl' میں global ویریبلس کے بارے میں سیکھیں گے۔
07:38	ایک global variable پروگرام میں کہیں بھی حاصل کیا جاسکتا ہے۔

07:43	'گلوبل ویریبلس 'our' کی وارڈ کے ساتھ ڈکلیئر ہوتے ہیں۔
07:47	یہاں کچھ مثالیں ہیں۔ our \$fvalue = 100 semicolon </nowiki> our \$fname =within double quotes Priya semicolon
08:01	ab، global ویریبلس کی ایک فعال مثال دیکھتے ہیں۔
08:06	'ٹرمنل' پرواپس جائیں اور ٹائپ کریں : 'gedit scope' ہافن our ڈاٹ 'pl ampersand' اور اینٹر دبائیں۔
08:16	یہ 'gedit' میں 'pl scope hyphen our' ظاہر کرے گا
08:22	ab میں وہ سیمپل پروگرام سمجھاتا ہوں جو میں نے لکھا ہے۔
08:27	میں نے 'package main' اور 'our \$ i' کی طرح ایک 'گلوبل ویریبل' ڈکلیئر کیا ہے اور اسے '100' سے انشلا نز کیا ہے۔
08:37	'package First' کے ڈکلیئر ہونے پر توجہ دیں۔
08:40	'package' کوڈ کا مجموعہ ہے جس کا خود کا 'namespace' ہوتا ہے۔
08:46	'Namespace' پیکیجیس کے درمیان variable name collisions کو بچاتا ہے۔
08:51	ہم 'package' اور 'namespace' کے بارے میں مزید آگے کے ٹیوٹوریلز میں دیکھیں گے۔
08:56	'package First' میں، گلوبل ویریبل 'i' ویلیو 10 رکھتا ہے۔
09:02	'package Second' میں گلوبل ویریبل 'i' کو ویلیو 20 تفویض کی جاتی ہے۔
09:08	مین پیکیج، 'package First variable' اور 'package Second variable' دونوں کو استعمال کرتا ہے۔
09:15	پروگرام میں، میں نے سارے packages میں ایک ہی ویریبل 'i' ڈکلیئر کیا ہے۔
09:21	package variable ، package name colon colon variable name سے بیان کردہ ہوتا ہے۔
09:29	ہماری مثال میں، یہ i colon colon \$Second، i colon colon \$First ہے۔
09:39	ہمارے پاس ایک فائل میں مختلف پیکیجیس ہیں اور گلوبل ویریبل تک تمام packages کی رسائی ہوتی ہے۔

09:47	اب، فائل کو سیو کریں اور پروگرام کو ایکزیکوٹ کریں۔
09:51	لہذا، terminal پر جائیں اور ٹائپ کریں : 'perl scope hyphen our dot pl' اور اینٹر دبائیں۔
09:59	terminal پر آؤٹ پٹ ظاہر ہوتا ہے۔
10:03	'ویریبل' i کی تفویض سمجھنے کے لئے آپ آؤٹ پٹ کا اپنے آپ تجزیہ کریں
10:11	اب یہ ٹیوٹوریل ختم ہوتا ہے۔ اس کا خلاصہ کرتے ہیں۔
10:16	اس ٹیوٹوریل میں، ہم نے مندرجہ ذیل کرنا سیکھا: مثالوں کے ساتھ ویریبلز کا سکوپ 'private variables' ڈیکلئر کرنا 'dynamically scoped variables' اور 'global variables'
10:29	کمپائلیشن تیزی سے کرنے کے لئے local کے بجائے 'my' استعمال کرنے کی صلاح دی جاتی ہے۔
10:35	یہاں آپ کے لئے ایک مشق ہے۔
10:37	مندرجہ ذیل مشق کے لئے کوڈ لکھیں اور اسے ایکزیکوٹ کریں۔
10:42	'FirstModule' کے طور پر ایک 'پیکج' کو ڈیکلئر کریں۔
10:46	ایک ویریبل '\$ age' کو 'our' کے طور پر ڈیکلئر کریں اور ویلیو '42' تفویض کریں۔
10:52	'SecondModule' کی طرح ایک دوسرا پیکج ڈیکلئر کریں۔
10:56	ایک ویریبل '\$ ageword' کو 'our' کے طور پر ڈیکلئر کریں اور ڈبل کوٹس میں "Forty-Two" ویلیو تفویض کریں۔
11:05	ایک سب روٹین 'First ()' ڈیکلئر کریں۔
11:08	سب-روٹین میں، مندرجہ ذیل کی طرح 'local' اور 'my' کی وارڈس کے ساتھ دو ویریبلز ڈیکلئر کریں:
11:16	'local' '\$ age = 52' semicolon
11:20	'\$ ageword' = 'my' ڈبل کوٹس میں Fifty-two semicolon
11:27	ایک اور سب-روٹین 'Result ()' کال کریں۔
11:31	اس فنکشن میں '\$ age' اور '\$ ageword' کی ویلیوز کو پرنٹ کریں۔

11:37	سب-روٹین کا اختتام کریں۔
11:39	سب-روٹین 'Result ()' کوڈ کلئیر کریں۔
11:42	'\$ age' اور '\$ ageword' کی ویلیوز کو دوبارہ پرنٹ کریں۔
11:47	سب-روٹین کا اختتام کریں۔
11:49	فنکشن 'First ()' کو کال کریں۔
11:51	نیچے کی طرح 'Package First' اور 'Package Second' کو پرنٹ کریں۔
11:57	مندرجہ ذیل لنکس پر دستیاب ویڈیو اسپوکن ٹیوٹوریل پروجیکٹ کا خلاصہ کرتا ہے۔ اسے ڈاؤن لوڈ کریں اور دیکھیں۔
12:05	اسپوکن ٹیوٹوریل پروجیکٹ ٹیم اسپوکن ٹیوٹوریلز کا استعمال کرتے ہوئے ورکشاپ چلاتی ہے اور آن لائن ٹیسٹ پاس کرنے والوں کو سند دیتی ہے۔ مزید معلومات کے لئے ہمیں لکھیں۔
12:18	اسپوکن ٹیوٹوریل پراجیکٹ بھارتی حکومت کے ایم ایچ آر ڈی کے NMEICT طرف حمایت شدہ ہے۔ اس مشن پر مزید معلومات اس لنک پر دستیاب ہیں۔
12:31	آئی آئی ٹی بابے سے میں وجاحت احمد آپ سے رخصت لیتا ہوں۔ شامل ہونے کے لئے شکریہ۔